



BATEC CONNECTED

Una Tesis de Grado
Sometida a la Facultad de
Escola Tècnica d'Enginyeria de Telecomunicació de
Barcelona
Universitat Politècnica de Catalunya
por
Mario Cahiz Sánchez

En cumplimiento parcial
de los requisitos para el grado en
INGENIERÍA DE SISTEMAS ELECTRÓNICOS

Tutor: Vicente Jiménez Serres

Barcelona, Enero 2016

Abstract

This thesis project is based on the integration of a data collection system with multiple sensors to be displayed in real time on a mobile application for the products of Batec Mobility S.L. in order to improve the customer experience.

For the project implementation hybrid mobile application programming was studied and a simple, easy to use and accessible application for people with motor disabilities oriented to the Batec's market was implemented. For the sensor system a prototype based on Arduino with Bluetooth connection to send data was implemented.

Once the project was done, in conclusion, we can state that it is a device whose final version will be a viable product for industrialization, implementation and subsequent launch in the Batec's current market as an innovation of the base product.

Resum

Aquest projecte de tesi es basa en la integració d'un sistema de recol·lecció de dades amb múltiples sensors sent visualitzats en temps real en una aplicació mòbil per als productes de l'empresa Batec Mobility S.L. amb la finalitat de millorar l'experiència dels seus clients.

Per a l'execució del projecte es va estudiar la programació d'aplicacions mòbils híbrides i es va implementar una aplicació senzilla, fàcil d'usar i accessible per a persones amb discapacitat motriu orientat al mercat de Batec. Per al sistema de sensors es va implementar un prototip basat en Arduino amb connexió Bluetooth per enviar les dades.

Un cop realitzat el projecte, en conclusió, podem afirmar que es tracta d'un dispositiu que en la seva versió final serà un producte viable per a la seva industrialització, implementació i posterior llançament al mercat actual de Batec com una millora d'innovació del producte base.

Resumen

Este proyecto de tesis se basó en la en la integración de un sistema de recolección de datos con múltiples sensores siendo visualizados en tiempo real en una aplicación móvil para los productos de la empresa Batec Mobility S.L. con el fin de mejorar la experiencia de sus clientes.

Para la ejecución del proyecto se estudió la programación de aplicaciones móviles híbridas y se implementó una aplicación sencilla, fácil de usar y accesible para personas con discapacidad motriz orientado al mercado de Batec. Para el sistema de sensores se implementó un prototipo basado en Arduino con conexión Bluetooth para enviar los datos.

Una vez realizado el proyecto, en conclusión, podemos afirmar que se trata de un dispositivo cuya versión final será un producto viable para su industrialización, implementación y posterior lanzamiento en el mercado actual de Batec como una mejora de innovación del producto base.

Dedicado a mis padres y a mi hermana por ser los pilares fundamentales en todo lo que soy y apoyarme tanto en los buenos como en los malos, que ha habido unos cuantos.

Todo este trabajo ha sido posible gracias a ellos. Simplemente gracias.

Agradecimientos

Primeramente, me gustaría agradecer a la empresa Batec Mobility S.L. toda la confianza y recursos que han depositado en este proyecto. A mi profesor Vicente Jiménez que me ha guiado perfectamente para la consecución de este propósito. A mi amigo Alex Masnou por introducirme en el mundo de la programación de las aplicaciones móviles. A mis compañeros de Fourier que han hecho esta difícil carrera mucho más llevadera. Y sobre todo a mi familia por apoyarme en todas las decisiones que he tomado.

Historial de revisiones y registro de aprobaciones

Revisión	Fecha	Propósito
0	03/01/2016	Creación del documento
1	22/01/2016	Revisión del documento

Lista de Participantes

Nombre	e-mail
Mario Cahiz Sánchez	mario.cahiz@gmail.com
Vicente Jiménez Serres	vicente.jimenez@upc.edu

Escrito por: Mario Cahiz Sánchez		Revisado y Aprobado por: Vicente Jiménez	
Fecha	03/01/2016	Fecha	22/01/2016
Nombre	Mario Cahiz Sánchez	Nombre	Vicente Jiménez Serres
Posición	Autor del Proyecto	Posición	Supervisor del Proyecto

Tabla de contenidos

Abstract	1
Resum	2
Resumen	3
Agradecimientos.....	5
Historial de revisiones y registro de aprobaciones.....	6
Tabla de contenidos	7
Lista de ilustraciones	9
Lista de Tablas	10
1. Introducción	11
1.1. Planificación y línea de tiempo.....	12
1.2. Hitos del proyecto.....	14
1.3. Diagrama de Gantt.	15
2. Tecnologías usadas o aplicadas en esta tesis.....	16
2.1. Batec Handbikes.	16
2.2. Ionic Framework.	16
2.3. Arduino.	17
2.4. Bluetooth.	18
3. Metodología / desarrollo del proyecto	19
3.1. Desarrollo del Software.	19
3.1.1. Selección y aprendizaje de la codificación de la aplicación Batec Connected.	19
3.1.2. Diseño básico de la aplicación.	20
3.1.3. Programación de la aplicación Batec Connected.	20
3.2. Desarrollo del Hardware.....	20
3.2.1. Selección de los parámetros a monitorizar con los sensores.	20
3.2.2. Selección del microprocesador.	21
3.2.3. Selección del tipo de comunicación a emplear.	21
3.2.4. Diseño e implementación de los circuitos de sensores escogidos.	21
3.2.5. Diseño del circuito de monitoreo de la tensión de la batería.	23
3.2.6. Diseño e implementación del circuito de comunicación serie con Bluetooth.....	25
3.2.7. Integración de todo el sistema.	26
3.2.8. Programación del Hardware en Arduino.	27
4. Resultados.....	29
4.1. Batec Connected: Home Page.....	29
4.2. Batec Connected: General Page.....	29
4.3. Batec Connected: Velocimeter Page.	30

4.4.	Batec Connected: Battery Page.	31
4.5.	Batec Connected: Configuration Page.	32
4.6.	Funciones de utilidad.....	32
5.	Presupuesto	34
6.	Desarrollo futuro y conclusiones	36
6.1.	Mejoras de Software.....	36
6.2.	Mejoras de Hardware.	36
	Bibliografía.....	38
	Glosario	39

Lista de ilustraciones

Ilustración 1. Diagrama de Gantt.....	15
Ilustración 2. Lista de tareas	15
Ilustración 3. Logotipo de Batec Mobility.....	16
Ilustración 4. Logotipo de Ionic Framework.	17
Ilustración 5. Logotipo de Arduino.....	18
Ilustración 6. Logotipo de Bluetooth.	18
Ilustración 7. Diseño del circuito de sensores en la plataforma Arduino.	22
Ilustración 8. Curva de descarga de la batería. En línea punteada, tenemos la aproximación.....	23
Ilustración 9. Diseño del circuito de la batería.	24
Ilustración 10. Implementación del circuito de la batería.	24
Ilustración 11. Curva de Respuesta del Circuito de la Batería.....	24
Ilustración 12. Conexión Bluetooth serie con Arduino.	25
Ilustración 13. Esquemático del circuito final.	26
Ilustración 14. Diseño del circuito final en la plataforma Arduino.	26
Ilustración 15. Batec Connected Home Page	29
Ilustración 16. Batec Connected General Page	30
Ilustración 17. Batec Connected Velocimeter Page	31
Ilustración 18. Batec Connected Battery Page.	31
Ilustración 19. Batec Connected Configuration Page.	32
Ilustración 20. Distribución del presupuesto de Batec Connected.....	35

Lista de Tablas

Tabla 1. Paquete de trabajo #1	12
Tabla 2. Paquete de trabajo #2	12
Tabla 3. Paquete de trabajo #3	13
Tabla 4. Paquete de trabajo #4	13
Tabla 5. Paquete de trabajo #5	13
Tabla 6. Paquete de trabajo #6	14
Tabla 7. Tabla de hitos del proyecto	14
Tabla 8. Presupuesto Batec Connected.	34

1. Introducción

Este proyecto ha sido desarrollado en el departamento de ingeniería electrónica de la UPC juntamente con la colaboración de la empresa Batec Mobility S.L., la cual diseña, fabrica y distribuye *handbikes* adaptables para usuarios con silla de ruedas.

La idea principal del proyecto surgió mientras estaba trabajando en la empresa Batec Mobility S.L. en el departamento de I+D, donde empecé a trabajar. Este departamento trabaja constantemente en la innovación, con el fin de desarrollar nuevas mejoras en el producto. Como estudiante de telecomunicaciones, me encontré con la idea de ser capaz de conectar dicho producto, el Batec, con cualquier tipo de dispositivo portátil que el usuario casi siempre trae consigo. Esto significaría que el Batec sería capaz de interactuar con el usuario y viceversa, mejorando su experiencia y satisfacción.

El proyecto consiste en la integración de un sistema recolector de datos para los productos de la empresa Batec Mobility S.L, mediante el uso de sensores y una aplicación móvil propia para dicha empresa móvil que permitirá visualizar dichos datos en tiempo real y consultarlos una vez el dispositivo móvil no se encuentre enlazado con el producto de la empresa.

El proyecto se estructura en tres partes claramente diferenciadas: Un sistema electrónico de captación de información; la aplicación móvil propia para Batec Mobility S.L.; y la interconexión entre ambos sistemas.

La primera parte consiste en la ejecución del sistema electrónico que supondrá la elección de un microprocesador adecuado y de diversos sensores externos para obtener la información deseada, cómo, por ejemplo, velocidad, estado de la batería y otros datos de importancia para el usuario del producto de Batec.

La segunda parte consiste en el diseño y la creación de una aplicación para móviles propia para Batec Mobility. Dicha aplicación no solo será útil para el propio proyecto de visualizar los datos obtenidos por el sistema de sensores, sino que también se pretende que en un futuro cercano sea una herramienta informativa y de marketing útil tanto para los usuarios de Batec como para la propia empresa.

Finalmente, la última parte consistirá en la interconexión de las dos partes anteriormente mencionadas, utilizando el protocolo de comunicaciones Bluetooth, para dispositivos móviles que requieren un corto alcance de emisión.

Una vez presentado el proyecto, los principales objetivos de este trabajo son los siguientes:

1. Mejorar el *handbike* Batec añadiendo el sistema ICT explicado anteriormente.
2. Obtener el conocimiento para diseñar e implementar aplicaciones móviles compatibles diferentes sistemas operativos que se encuentran en el mercado.
3. Crear una aplicación propia para la empresa Batec Mobility S.L. teniendo en cuenta que dicha aplicación debe ser accesible para todas las posibles limitaciones físicas que los usuarios de Batec tengan.
4. Diseñar, construir e implementar un circuito de múltiples sensores capaz de comunicarse sin cables con una aplicación móvil en un terminal telefónico.

Este es un proyecto independiente, sin embargo, se llevará a cabo en el marco de la empresa Batec Mobility S.L. debido a su implicación directa con el producto base. La empresa me ha autorizado a utilizar sus productos con el fin de desarrollar este proyecto correctamente y tener más posibilidades de éxito.

1.1. Planificación y línea de tiempo

Seguidamente podemos ver la planificación del proyecto que se llevó a cabo, vista a través de seis diferentes paquetes de trabajo. En estos paquetes de trabajo se muestra la actividad principal a realizar, con una descripción corta; la fecha en que se planeó empezar y acabar el bloque de trabajo juntamente con la real; una pequeña lista de tareas internas específicas de cada bloque y, finalmente, se detallan una serie de entregas al finalizar cada paquete de trabajo, en el caso que proceda.

Proyecto: Batec Connected	PT ref. PT#1	
Actividad principal: Aprendizaje del Software	Hoja 1 of 3	
Breve descripción: El aprendizaje la lenguajes de programación de las aplicaciones para <i>Smartphones</i> con el fin de implementar una única aplicación capaz de funcionar en diferentes plataformas móviles.	Fecha inicio prevista: 15-09-2015 Fecha fin prevista: 26-10-2015	
	Inicio evento: 15-09-2015 Fin evento: 09-11-2015	
Tarea interna T1: - Aprendizaje de HTML5. Tarea interna T2: - Aprendizaje de CSS3. Tarea interna T3: - Aprendizaje de AngularJS.	Entregables:	Fechas:
	-	-

Tabla 1. Paquete de trabajo #1

Proyecto: Batec Connected	PT ref. PT#2	
Actividad principal: Programación del Software	Hoja 1 of 3	
Breve descripción: La creación de una aplicación móvil para Batec Mobility S.L., que sea capaz de integrarse en los productos de la compañía.	Fecha inicio prevista: 27-10-2015 Fecha fin prevista: 18-12-2015	
	Inicio evento: 10-11-2015 Fin evento: 23-12-2015	
Tarea interna T1: - Diseño del Concepto de la Aplicación. Tarea interna T2: - Workflow de la Aplicación. Tarea interna T3: - Funcionamiento de la Aplicación.	Entregables:	Fechas:
	Revisión Crítica	18-12-2015

Tabla 2. Paquete de trabajo #2

Proyecto: Batec Connected	PT ref. PT#3	
Actividad principal: Diseño de la Electrónica	Hoja 2 of 3	
Breve descripción: Diseño y selección de los componentes electrónicos adecuados para implementar la circuitería básica para entregar datos para la aplicación móvil.	Fecha inicio prevista: 21-09-2015 Fecha fin prevista: 09-10-2015	
	Inicio evento: 12-10-2015 Fin evento: 30-10-2015	
Tarea interna T1: - Seleccionar el Microprocesador. Tarea interna T2: - Seleccionar los componentes y los sensores. Tarea interna T3: - Seleccionar el módulo de comunicación.	Entregables:	Fechas:
	-	-

Tabla 3. Paquete de trabajo #3

Proyecto: Batec Connected	PT ref. PT#4	
Actividad principal: Implementación de Electrónica	Hoja 2 of 3	
Breve descripción: Implementación de la circuitería electrónica con todos los sensores funcionando correctamente y siendo capaz de transmitir y recibir datos con un protocolo de conexión inalámbrica.	Fecha inicio prevista: 12-10-2015 Fecha fin prevista: 18-12-2015	
	Inicio evento: 02-11-2015 Fin evento: 23-12-2015	
Tarea interna T1: - Construir el Circuito Electrónico. Tarea interna T2: - Programar el Microprocesador. Tarea interna T3: - Testear los Sensores individualmente.	Entregables:	Fechas:
	Revisión Crítica	18-12-2015

Tabla 4. Paquete de trabajo #4

Proyecto: Batec Connected	PT ref. PT#5	
Actividad principal: Integración del Sistema	Hoja 2 of 3	
Breve descripción: Integración de la parte electrónica con la propia aplicación móvil, estableciendo una conexión segura.	Fecha inicio prevista: 21-12-2015 Fecha fin prevista: 15-01-2016	
	Inicio evento: 24-12-2015 Fin evento: 15-01-2016	
	Entregables:	Fechas:
	Reunión de progreso 2	15-01-2016

Tabla 5. Paquete de trabajo #5

Proyecto: Batec Connected	PT ref. PT#6	
Actividad principal: Test Final	Hoja 3 of 3	
Breve descripción: Testeo de todo el sistema con el fin de demostrar que se cumplan todos los requisitos.	Fecha inicio prevista: 18-01-2016 Fecha fin prevista: 22-01-2016	
	Inicio evento: 18-01-2016 Fin evento: 22-01-2016	
	Entregables: Revisión Final	Fechas: 22-01-2016

Tabla 6. Paquete de trabajo #6

1.2. Hitos del proyecto.

En la siguiente tabla podemos ver la lista de hitos del proyecto con su fecha de finalización planificada.

WP#	Tarea#	Título corto	Hito / Entregable	Fecha
1	-	Aprendizaje del Software		26-10-2015
	1	HTML5	-	28-09-2015
	2	CSS3	-	12-10-2015
	3	AngularJS	-	26-10-2015
2	-	Programación de la Aplicación		18-12-2015
	1	Diseñar el Concepto de la App	-	02-11-2015
	2	Workflow de la Aplicación	Reunión de Progreso 1	11-11-2015
	3	Funcionamiento de la Aplicación	Revisión Crítica	18-12-2015
3	-	Diseño del Circuito		09-10-2015
	1	Seleccionar el Microprocesador	-	25-09-2015
	2	Seleccionar los Componentes	-	02-10-2015
	3	Seleccionar el modulo inalámbrico	-	09-10-2015
4	-	Implementar el Circuito		18-12-2015
	1	Construir el Circuito	Reunión de Progreso 1	06-11-2015
	2	Programar el Microprocesador	-	04-12-2015
	3	Testear los Sensores	Revisión Crítica	18-12-2015
5	-	Integrar el Sistema	Reunión de Progreso 2	15-01-2016
6	-	Test Final	Revisión Final	22-01-2016

Tabla 7. Tabla de hitos del proyecto

1.3. Diagrama de Gantt.

Debido a las dificultades en el aprendizaje de los tres lenguajes de programación, HTML5, CSS3 y AngularJS, el proyecto experimentó un retraso en la parte electrónica, ya que, en este punto se quería centrar los esfuerzos en terminar la parte principal de la aplicación. A pesar de este retraso inesperado, la parte electrónica del proyecto se puso al día con el plan inicial.

En las dos siguientes ilustraciones podemos ver el diagrama de Gantt final del proyecto juntamente con la lista de tareas asociada.

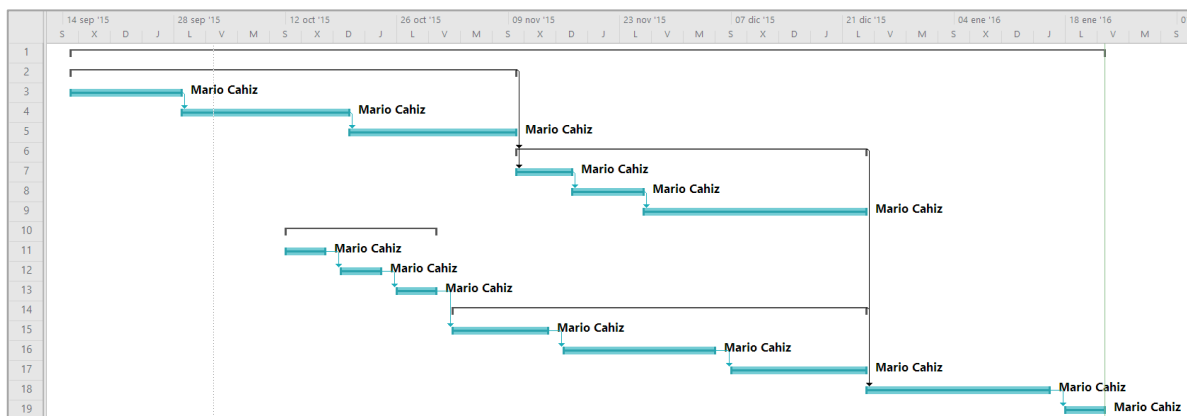


Ilustración 1. Diagrama de Gantt

	i	Modi de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	✓	📁	▸ Batec Connected	94 días	mar 15/09/15	vie 22/01/16		
2	✓	📁	▸ Aprendizaje del Software	40 días	mar 15/09/15	lun 09/11/15		
3	✓	📌	HTML5	10 días	mar 15/09/15	lun 28/09/15		Mario Cahiz
4	✓	📌	CSS3	15 días	mar 29/09/15	lun 19/10/15	3	Mario Cahiz
5	✓	📌	AngularJS	15 días	mar 20/10/15	lun 09/11/15	4	Mario Cahiz
6	✓	📁	▸ Programación de la Aplicación	32 días	mar 10/11/15	mié 23/12/15	2	
7	✓	📌	Diseñar el Concepto de la App	5 días	mar 10/11/15	lun 16/11/15	2	Mario Cahiz
8	✓	📌	Workflow de la Aplicación	7 días	mar 17/11/15	mié 25/11/15	7	Mario Cahiz
9	✓	📌	Funcionamiento de la Aplicación	20 días	jue 26/11/15	mié 23/12/15	8	Mario Cahiz
10	✓	📁	▸ Diseño del Circuito	15 días	lun 12/10/15	vie 30/10/15		
11	✓	📌	Seleccionar el Microprocesador	5 días	lun 12/10/15	vie 16/10/15		Mario Cahiz
12	✓	📌	Seleccionar los Componentes	5 días	lun 19/10/15	vie 23/10/15	11	Mario Cahiz
13	✓	📌	Seleccionar el módulo inalámbrico	5 días	lun 26/10/15	vie 30/10/15	12	Mario Cahiz
14	✓	📁	▸ Implementar el Circuito	38 días	lun 02/11/15	mié 23/12/15		
15	✓	📌	Construir el Circuito	10 días	lun 02/11/15	vie 13/11/15	13	Mario Cahiz
16	✓	📌	Programar el Microprocesador	15 días	lun 16/11/15	vie 04/12/15	15	Mario Cahiz
17	✓	📌	Testear los Sensores	13 días	lun 07/12/15	mié 23/12/15	16	Mario Cahiz
18	✓	📌	Integrar el Sistema	17 días	jue 24/12/15	vie 15/01/16	6;14	Mario Cahiz
19	✓	📌	Test Final	5 días	lun 18/01/16	vie 22/01/16	18	Mario Cahiz

Ilustración 2. Lista de tareas

2. Tecnologías usadas o aplicadas en esta tesis

En este apartado se va a dar una pequeña introducción sobre las tecnologías y los productos que forman parte de este proyecto.

2.1. Batec Handbikes.

Batec Mobility S.L. es una empresa que mejora radicalmente la movilidad de las personas con discapacidad física mediante el diseño, la fabricación y comercialización de *handbikes* acoplables. Un *handbike* o bicicleta de mano es un tipo de vehículo impulsado por los brazos en lugar de las piernas, como en una bicicleta. La mayoría de los *handbikes* son triciclos en la forma, con dos ruedas traseras y una rueda delantera accionada y dirigitiva.

Estos *handbikes* se pueden acoplar hasta en un 99% de las sillas de ruedas que hay actualmente en el mercado. Además, Batec Mobility S.L. tiene como objetivo crear empleo para personas con discapacidad física y predicar con el ejemplo al tener al 50% de la plantilla formada por personas con alguna discapacidad física.

Los *handbikes* Batec están divididos en tres modelos claramente diferenciados: el Batec Manual, el Batec Eléctrico y el Batec Híbrido. El Batec Manual es el modelo de *handbike* propulsado manualmente con los brazos, que resulta perfecto para hacer ejercicio y desplazarse. El Batec Eléctrico es el modelo de *handbike* que incorpora un motor en su rueda delantera para un desplazamiento más cómodo e ideal para el uso diario. Por último, el Batec Híbrido es la combinación de los dos modelos anteriores, con tracción manual, pero con la asistencia de un motor en su rueda delantera. Cada modelo se adapta a las necesidades de cada uno de los usuarios con una gran variedad de adaptaciones disponibles que ayudan al usuario a poder llevar uno de estos vehículos.

Batec Mobility S.L. fue fundada en 2006 por su creador Pau Bach, un gran apasionado de las motos y las bicicletas, que pocos años antes sufrió un accidente de tráfico que le dejó tetrapléjico.



Ilustración 3. Logotipo de Batec Mobility.

2.2. Ionic Framework.

Ionic® es un completo SDK para el desarrollo de aplicaciones móviles híbridas basadas en HTML5, CSS3 y JavaScript. Está construido con SASS y optimizado con AngularJS. Las aplicaciones híbridas son una combinación entre una aplicación web y una aplicación nativa. Esta combinación permite mantener la gran ventaja de cada una de las aplicaciones anteriores. Por un parte, conservan el carácter multiplataforma de las aplicaciones web y por otra, al correr localmente en el dispositivo pueden ejecutarse sin conexión a Internet y, además al estar embebidas en el dispositivo móvil, tienen el mismo

tipo de acceso a las APIs nativas de cada sistema operativo, así como a los recursos propios del sistema tipo procesador, GPS, cámara, etc.

Ionic está basado en la ideología de separación de conceptos MVC (*Model-View-Controller*), que como su nombre indica, está separado en tres componentes: Modelo, Vista y Controlador.

El modelo es la capa encargada de los datos, es decir, la que se encarga de hacer peticiones a las bases de datos para enviar o recibir información. Estas bases de datos pueden estar alojadas de forma local en nuestra app o de forma remota en un servidor externo.

La Vista se trata del código que nos permitirá presentar los datos que el modelo nos proporciona, como ejemplo podríamos decir que en una aplicación es el código HTML que nos permite mostrar la salida de los datos procesados.

Finalmente, el controlador es la capa que sirve de enlace entre la vista y el modelo. Envía comandos al modelo para actualizar su estado, y a la vista correspondiente para cambiar su presentación.

Ionic utiliza el AngularJS con el fin de crear un marco más adecuado para desarrollar aplicaciones ricas y robustas. Ionic no sólo tiene una gran representación visual gracias al lenguaje de estilos SASS, sino que su arquitectura central es robusta y seria para el desarrollo de aplicaciones.

Ionic fue creado por Max Lych, Ben Sperry y Adam Bradley en 2013.



Ilustración 4. Logotipo de Ionic Framework.

2.3. Arduino.

Arduino® es una plataforma electrónica de código abierto para la creación de prototipos basados en hardware y software fácil de utilizar. Las placas de Arduino son sistemas que proporcionan conjuntos de pines digitales y analógicos de entrada y/o salida que pueden conectarse a varias placas de expansión o a otros circuitos y que están controlados por un microcontrolador, el cual puede ejecutar un código previamente cargado.

Para programar estos microcontroladores, Arduino ofrece un entorno de desarrollo integrado (IDE) multiplataforma que se programa utilizando un lenguaje propio basado en un lenguaje de programación de alto nivel muy similar a C++. Además, Arduino cuenta con una enorme comunidad de desarrollo, que tiene como anfitrión a GitHub, la cual es muy proactiva y en el foro hay una gran cantidad de proyectos y ejemplos con los que poder aprender rápidamente.

Arduino se inició en el año 2005 como un proyecto para estudiantes en el Instituto IVREA, en Ivrea (Italia).



Ilustración 5. Logotipo de Arduino.

2.4. Bluetooth.

Creado en 1994, la tecnología Bluetooth® fue concebida como una alternativa sin cables para intercambiar datos entre dos dispositivos usando transmisiones de radio. El nombre Bluetooth proviene del rey danés Harald Blåtand, o en inglés Harold Bluetooth. En el siglo XX, como dice la historia, el rey Blåtand ayudó a unir a las facciones en guerra de los países Nórdicos. De manera similar, la tecnología Bluetooth fue creada para permitir la conectividad y la colaboración entre diversos productos e industrias, como un estándar abierto para todos.

La tecnología Bluetooth es un protocolo de comunicaciones que trabaja en la banda de ISM de los 2.4 GHz. Está especialmente diseñado para dispositivos que requieren un consumo bajo, como es el caso de equipos portátiles dotados de una batería limitada, y que además requieran una conexión inalámbrica de corto alcance.

Desde su creación ha habido multitud de versiones, desde la inicial 1.0 hasta la actual 4.0. Con cada una de las versiones se ha ido mejorando la velocidad de transmisión de datos y el consumo por parte de los dispositivos que implementan este protocolo. Incluso existe una mejora de la versión 4.0, el denominado BLE o Bluetooth Low Energy, rebajando enormemente el consumo a expensas de suprimir la capacidad para transmitir voz.



Ilustración 6. Logotipo de Bluetooth.

3. Metodología / desarrollo del proyecto

Este capítulo se centrará en el desarrollo completo de todo el proyecto.

3.1. Desarrollo del Software.

El desarrollo del Software incluye el progreso de codificación de la aplicación para dispositivos móviles y el desarrollo del programa cargado en la plataforma Arduino.

3.1.1. Selección y aprendizaje de la codificación de la aplicación Batec Connected.

Antes de empezar a programar se tenía que tomar una decisión crucial que definiría el proyecto por completo, la elección del lenguaje o lenguajes adecuados para el diseño de la aplicación.

Ya que en mi caso no tenía experiencia en la creación de aplicaciones móviles, tuve que buscar un lenguaje de programación que cumpliera indispensablemente con dos requisitos. Uno era que la aplicación debía tener acceso a los recursos de hardware de los dispositivos móviles, como es el caso del acceso al Bluetooth del terminal para poder establecer un enlace de comunicaciones. Y la otra era que la aplicación debía abarcar los principales sistemas operativos móviles del mercado, es decir, un lenguaje multiplataforma. Con estas dos premisas investigué las diferentes alternativas que me recomendaron. Finalmente, la elección fue un SDK llamado Ionic Framework, el cual permite crear aplicaciones híbridas entre una aplicación web, que garantiza la compatibilidad con los diferentes sistemas operativos; y una aplicación nativa que, ofrece la posibilidad de acceder a la configuración del Hardware físico del dispositivo en cuestión. Además, Ionic presenta una gran variedad de animaciones y tiene una gama muy amplia de edición de estilos en su SDK.

Ionic trabaja en un entorno basado en tres lenguajes diferentes propios de las aplicaciones web: HTML5 para la creación de la estructura básica de la aplicación; CSS3 para la edición de estilos de la aplicación y AngularJS para la implementación de las funciones que se ejecutarán en la aplicación.

Debido a que no tenía conocimiento de los lenguajes de programación comentados anteriormente, aprendí estos lenguajes mediante diferentes métodos.

En el caso de HTML5, hice un curso online llamado *Introduction to HTML5* impartido por la universidad de Michigan de EEUU. Asimismo, complementé este curso con tutoriales online de estamentos como *World Wide Web Consortium (W3C)* y consultando publicaciones de páginas webs de creación de software libre como *Github*.

Para el aprendizaje de CSS3 también hice un curso online llamado *Introduction to CSS3* también impartido por la Universidad de Michigan. Además, complementé el curso con tutoriales online de W3C y consultando publicaciones de páginas webs de creación de software libre.

También para conocer el método de programación de AngularJS realicé otro curso online de la Universidad de Michigan llamado *Interacting with JavaScript*, y en este caso lo complementé con la guía oficial online de AngularJS.

Esta etapa de aprendizaje de los lenguajes de programación se prolongó demasiado, sufriendo así una demora en el global del proyecto.

3.1.2. Diseño básico de la aplicación.

Para el diseño de la aplicación me basé en dos consignas principales. La aplicación tenía que ser sencilla, fácil de usar y con una interfaz clara que permitiera que con un golpe de vista tuvieras la mayor información disponible.

La aplicación también debía ser accesible para personas con discapacidad motriz, especialmente aquellas personas que tienen dificultades en el movimiento libre de las manos. Para conseguir este objetivo, reduje lo máximo posible el *workflow* o interacciones para cambiar de pantalla, así con pocos movimientos se puede fácilmente cambiar la aplicación y, además así resulta más intuitiva para cualquier usuario.

Siguiendo estas directrices, decidí que la aplicación estaría estructurada en cinco páginas básicas, donde en cada una de ellas se mostrarían diferentes parámetros obtenidos mediante los sensores.

3.1.3. Programación de la aplicación Batec Connected.

Una vez diseñada la estructura y establecido el *workflow* que tendría la aplicación era el momento de empezar a programar la aplicación.

Primero se programó la estructura de las diferentes páginas de la aplicación con el lenguaje de codificación web HTML5. Se usaron diferentes tipos de etiquetas referentes a esta codificación como `<div>`, ``, `<button>`, entre otras, referentes a la HTML5 y otras como `<ion-slide>` o `<ion-radio>` que son exclusivas de Ionic Framework y tienen funciones y animaciones especiales.

Cuando la estructura de las diferentes páginas estuvo acabada, se procedió a la edición de estilos en CSS3 para hacer más atractiva y dinámica la aplicación.

Después de quedar satisfecho con el diseño de la aplicación ultimaba realizar todas las funciones con AngularJS.

Todos los archivos correspondientes al código desarrollado para programar la aplicación móvil se encuentran en el archivo comprimido adjunto a esta memoria con comentarios explicativos en cada función.

3.2. Desarrollo del Hardware.

En este apartado se presenta el desarrollo del Hardware utilizado en la consecución del prototipo del proyecto Batec Connected.

3.2.1. Selección de los parámetros a monitorizar con los sensores.

Primeramente, se debían seleccionar los parámetros que quería mostrar al usuario. Estos parámetros o magnitudes son los siguientes:

- Velocidad actual y media.
- Kilómetros recorridos totales y parciales.
- Tiempo parcial en funcionamiento.
- Estado del acelerador.
- Estado de carga de la batería.
- Consumo actual y medio.
- Temperatura ambiente.

Para medir y obtener los tres primeros parámetros de la lista decidí utilizar un sensor *reed* como el que utilizan los velocímetros de las bicicletas. Como Batec Mobility utiliza uno de estos sensores, Batec me proporcionó una unidad para poder utilizar en el prototipo.

Para obtener el estado del acelerador y el consumo, utilicé un recambio del acelerador utilizado en los *handbikes* Batec donde podía extraer el consumo en Amperios por la proporcionalidad de la cantidad de energía entregada al motor, también proporcionado por la empresa. Para medir el estado de carga de la batería se usaría la tensión en bornes de una batería suministrada por la empresa.

Finalmente, para obtener la temperatura ambiente se utilizó un sensor de temperatura que tuviera rango suficiente para poder registrar temperaturas bajo cero. Esta medida es importante para avisar al usuario que, con temperaturas excesivamente altas o bajas, la batería pierde eficacia.

3.2.2. Selección del microprocesador.

Para la selección del microprocesador, se barajaron varios procesadores, pero para la ejecución del prototipo se eligió Arduino ya que era un entorno de trabajo utilizado anteriormente para otros proyectos y tenía experiencia programando en su lenguaje. Además, el kit de desarrollo de Arduino incorpora una memoria EEPROM con una codificación muy sencilla, la cual era esencial para llevar a cabo dicho proyecto.

3.2.3. Selección del tipo de comunicación a emplear.

Una vez se eligieron los sensores y el microprocesador, era el momento de elegir qué tipo de enlace de comunicaciones emplearía para la conexión entre el microprocesador y la aplicación en el terminal móvil.

De entre todos los tipos de comunicaciones inalámbricas que podía utilizar en este proyecto me decanté por la comunicación Bluetooth a ser una tecnología estandarizada, integrada en la gran mayoría de dispositivos móviles, fácil de usar e integrable en ambos dispositivos de envío y recepción de datos utilizados.

Sin embargo, tenía que decidir entre los diferentes tipos de Bluetooth que hay, como el 2.0 o el nuevo BLE, o Bluetooth de bajo consumo. En mi caso, escogí la tecnología Bluetooth 2.0 porque la potencia de transmisión es más elevada en comparación con el Bluetooth LE y con éste último sería muy probable tener problemas de comunicación debido a la alta interferencia electromagnética que produce el motor del *handbike* Batec.

3.2.4. Diseño e implementación de los circuitos de sensores escogidos.

Seguidamente presentaremos el diseño y la implementación de los circuitos sensores escogidos. En este apartado se dará una explicación más funcional de cada uno de los circuitos presentes en el prototipo.

Sensor *reed*.

El sensor *reed* es un componente que actúa como un interruptor en función del campo magnético detectado. El sensor cierra el circuito cuando el campo magnético de un imán pasa cerca suyo.

En mi proyecto utilicé el sensor de *reed* de un odómetro, *SIGMA BC 5.12 Bike Computer*, suministrado por la empresa. El sensor se conecta directamente entre la alimentación y

una de las entradas de las entradas digitales que tiene Arduino, en la que se habilita un *pull-down*. Si el sensor no detecta ningún campo magnético cercano, el circuito permanece en circuito abierto y Arduino lo captura como un 0 lógico. En el instante en que detecta el campo electromagnético el sensor cierra el circuito y se eleva la tensión a 5V, por tanto, Arduino lo detecta como un 1 lógico. Entonces mediante software se obtiene el instante de tiempo en que el circuito marca dicha tensión y a partir de este dato ya puedo obtener los parámetros de velocidad, recorrido y tiempo del *handbike*.

Acelerador.

El acelerador, modelo *EZEE Twist Throttle*, es el equivalente a un potenciómetro cuyo valor resistivo varía según su posición. Para obtener el estado del acelerador obtuve la salida de la tensión de control del acelerador, que controla la energía entregada al motor, a través de una entrada analógica de Arduino. Después, el programa cargado en Arduino sintetiza su valor entre 0 y 100 para saber en qué porcentaje se encuentra el accionamiento del acelerador. Además, como el acelerador es el sensor que nos indica la energía entregada al motor y ésta es proporcional al consumo instantáneo del vehículo, podemos obtener la corriente de entrada en Amperios mediante la programación de Arduino.

Sensor temperatura.

El sensor de temperatura utilizado, modelo *MICROCHIP MCP 9700A-E/TO Thermistor*, se trata de un circuito integrado que varía su tensión de salida en función de la temperatura ambiente. La elección de este modelo en concreto se decidió porque podía registrar valores negativos de temperatura ambiente para zonas con un clima frío.

Para obtener la temperatura ambiente, monitorizamos la tensión de salida del circuito integrado a través de una entrada analógica de Arduino y el programa cargado en el dispositivo se encarga de obtener el valor de la temperatura ambiente en grados Celsius.

En la *Ilustración 7* podemos ver el diseño de los circuitos sensores explicados antes.

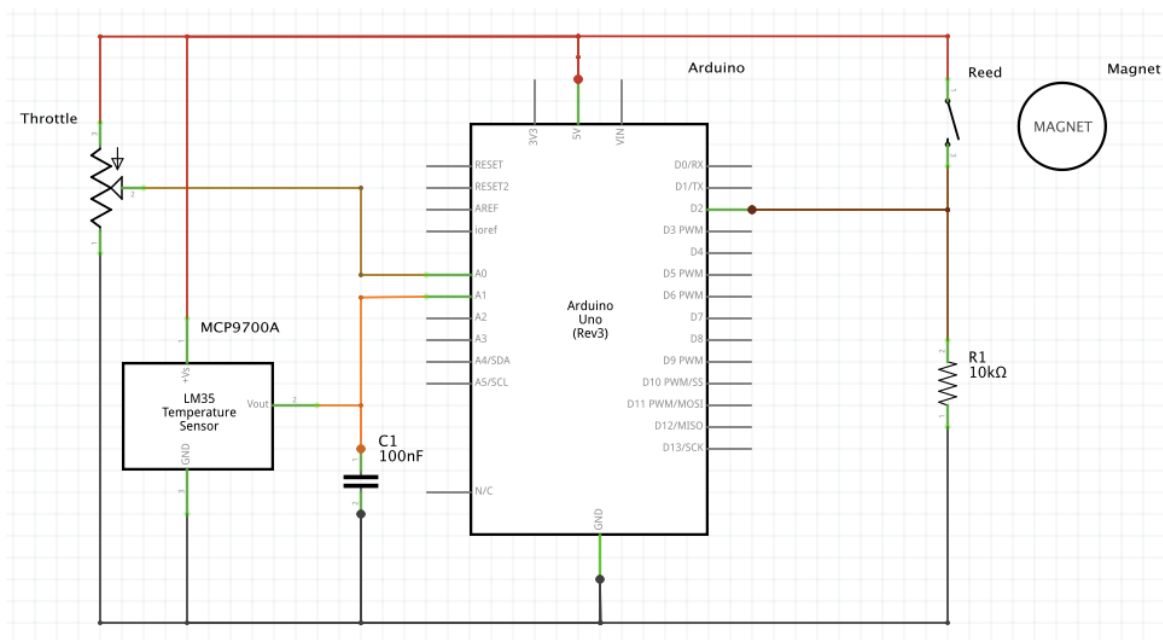


Ilustración 7. Diseño del circuito de sensores en la plataforma Arduino.

3.2.5. Diseño del circuito de monitoreo de la tensión de la batería.

Primeramente, para poder determinar el estado de la carga de la batería se tuvo que realizar un pequeño estudio de la batería utilizada por Batec Mobility S.L. en sus *handbikes*. Debido a que no fue posible encontrar el *datasheet* de la batería, se extrajo la curva de descarga de la batería a partir de la curva de descarga de cada una de las celdas que la componen en su respectivo *datasheet*. Esta curva se extrapoló a una curva de distribución equivalente a la cantidad de celdas en serie y paralelo para obtener los valores nominales de 36 V y 11 Ah de la batería. Posteriormente, se efectuó una aproximación lineal de orden 5 que representaba bastante fielmente la curva original, véase *Ilustración 8*.

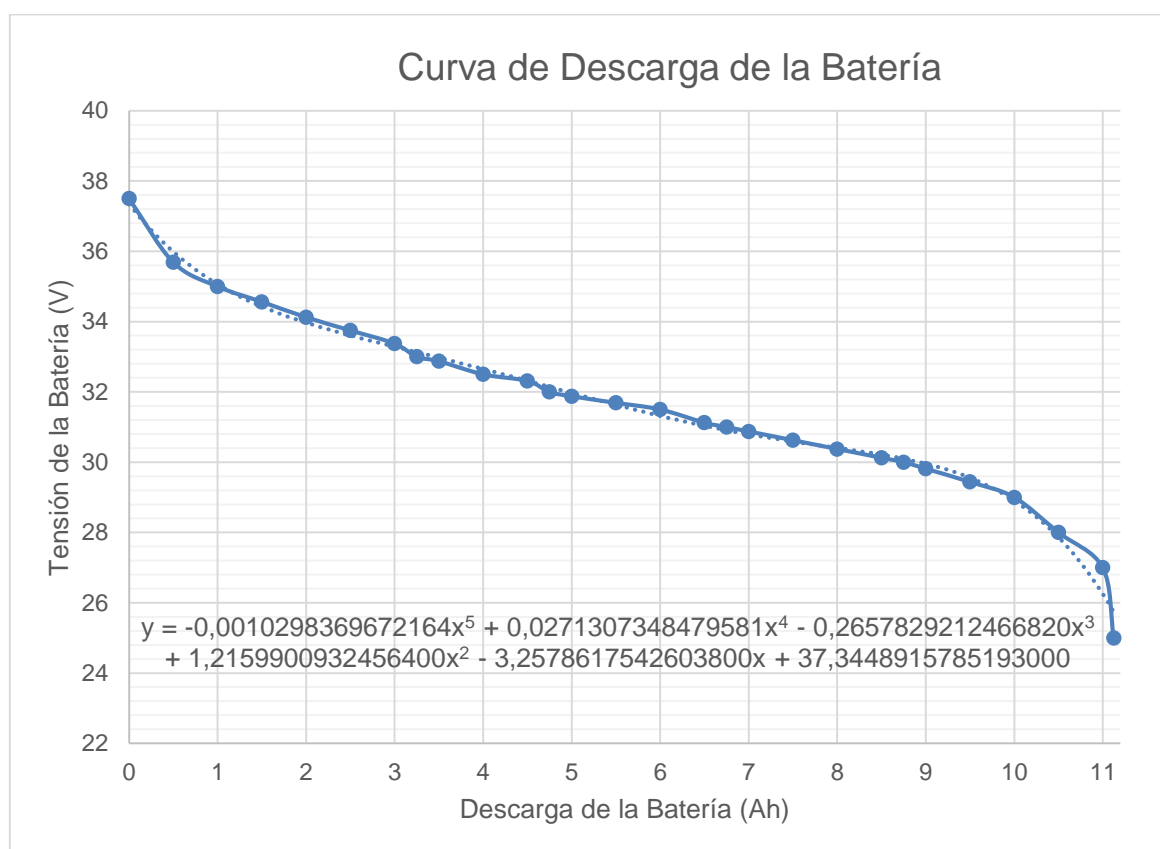


Ilustración 8. Curva de descarga de la batería. En línea punteada, tenemos la aproximación.

Una vez obtenida esta curva se implementó un circuito por el cual se reduce el rango de tensiones de la batería a un rango que permite operar con Arduino, es decir, de 5 a 0 V. Por este motivo, se diseñó un divisor de tensión con circuito de polarización y un diodo Zener de 25 V que efectúan la función anterior. En este circuito introducimos un transistor BJT para que el consumo de corriente del divisor de tensión sea cero cuando no se está midiendo la tensión de la batería. Este circuito está representado en la *Ilustración 9*.

Debido a una mayor disponibilidad de diodos zener de 12 V, implementamos el circuito de la *Ilustración 10*, el funcionamiento del cual es prácticamente idéntico al diseñado.

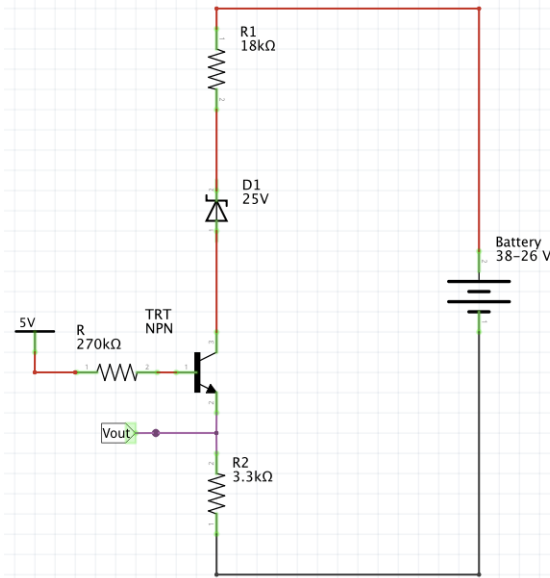


Ilustración 9. Diseño del circuito de la batería.

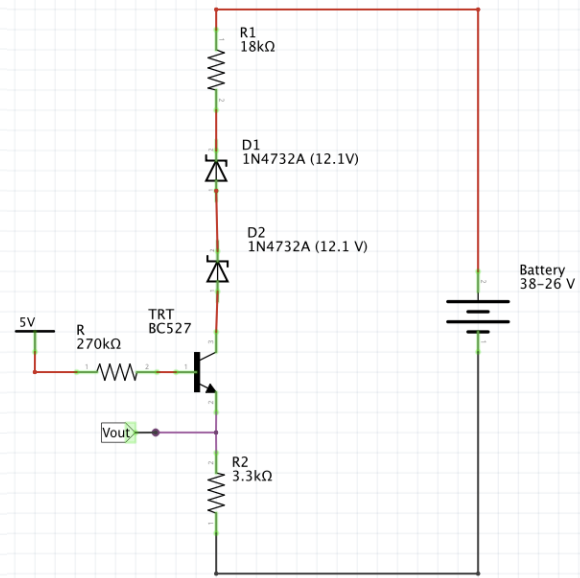


Ilustración 10. Implementación del circuito de la batería.

En el circuito anterior, primeramente, polarizamos el transistor con una fuente de tensión de 5 voltios, que proviene de Arduino, para asegurarnos que el transistor trabaja en zona de saturación y poder obtener la tensión de la batería. Seguidamente, los dos diodos Zener en serie reducen 24,2 V la tensión de entrada y el divisor de tensión resistivo nos la reduce también en un factor de 0,1549. Así conseguimos reducir el rango de tensiones de 38-26 V a un rango de 0,5-2,5 V, tal y como se puede ver en la curva de respuesta real del circuito, *Ilustración 11*.

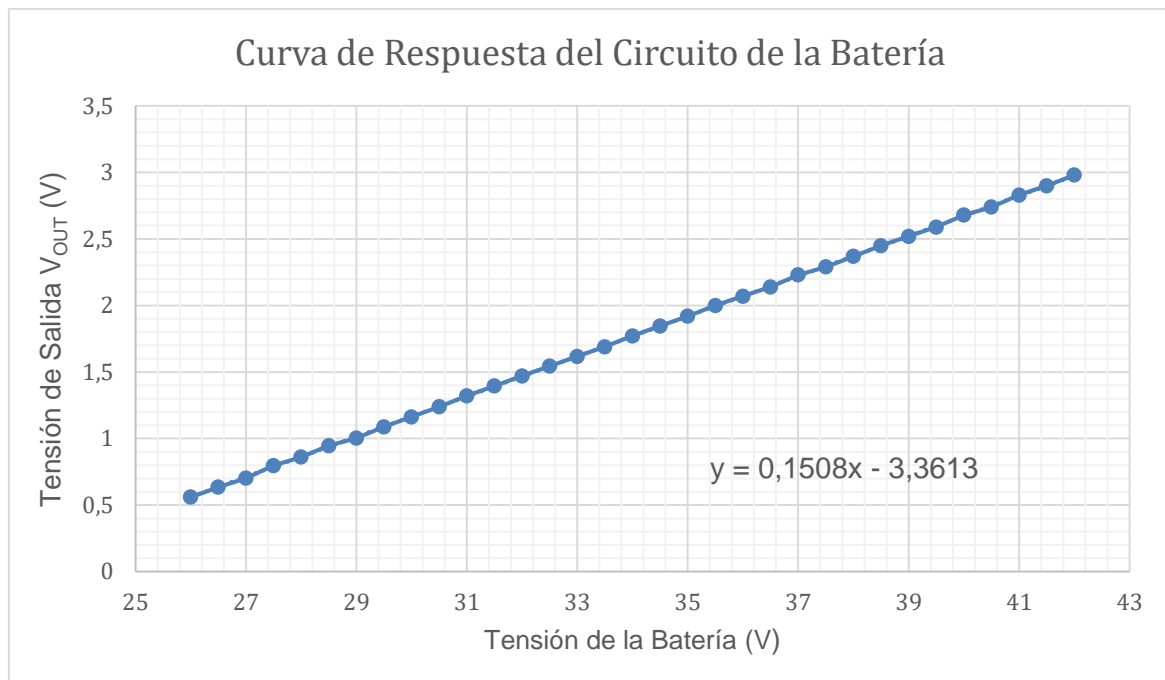


Ilustración 11. Curva de Respuesta del Circuito de la Batería

3.2.6. Diseño e implementación del circuito de comunicación serie con Bluetooth.

Una vez realizado los circuitos de sensores, el siguiente paso era la transmisión de los datos adquiridos por esos sensores a través de una comunicación serie Bluetooth. Para ello, se escogió el módulo *MICROCHIP RN42XVP-I/RM BT*, el cual se ha conectado, a nivel de prototipo, con una pequeña placa que hacía mucho más accesibles las conexiones del módulo.

Para poder realizar la conexión serie simplemente se tenía que conectar el pin de transmisión del Arduino con el pin de recepción del módulo y viceversa. No obstante, debido a que el módulo trabaja con tensiones alrededor de 3,3 V fue necesario realizar un divisor de tensión en la conexión entre la transmisión del Arduino, que se encuentra en el rango de 0 a 5 V, y el módulo Bluetooth para no dañar el módulo. También se corroboró que la señalización de 3,3 V del módulo Bluetooth en la transmisión es suficientemente alta para ser interpretada correctamente por Arduino. Entonces el diseño final se puede ver en la *Ilustración 12*.

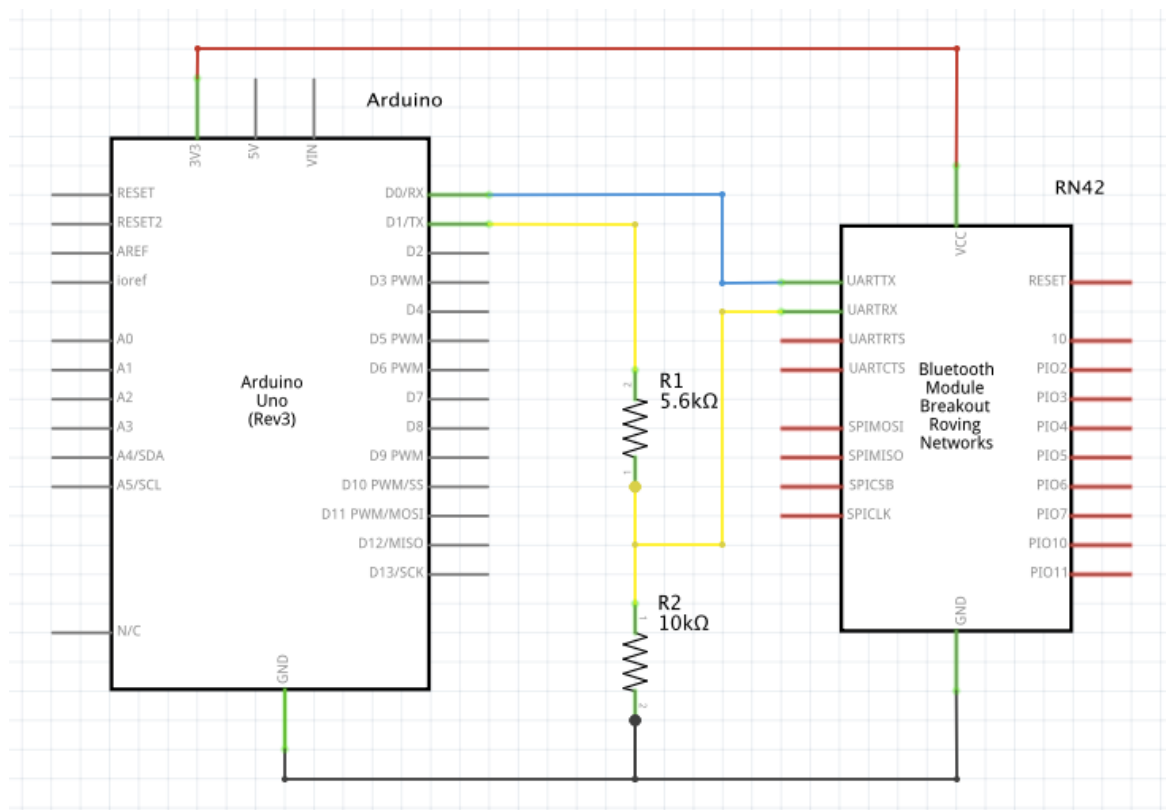


Ilustración 12. Conexión Bluetooth serie con Arduino.

3.2.7. Integración de todo el sistema.

Finalmente, la última parte era la implementación de los circuitos sensores y del circuito de comunicaciones Bluetooth. Una vez comprobado el funcionamiento de cada circuito individualmente, era el momento de probar todo el sistema junto. En la *Ilustración 13*, podemos ver el esquemático completo del Hardware implementado en el proyecto. Debe especificarse, que en la versión final del prototipo la resistencia de base del transistor será controlada por una salida digital del MCU.

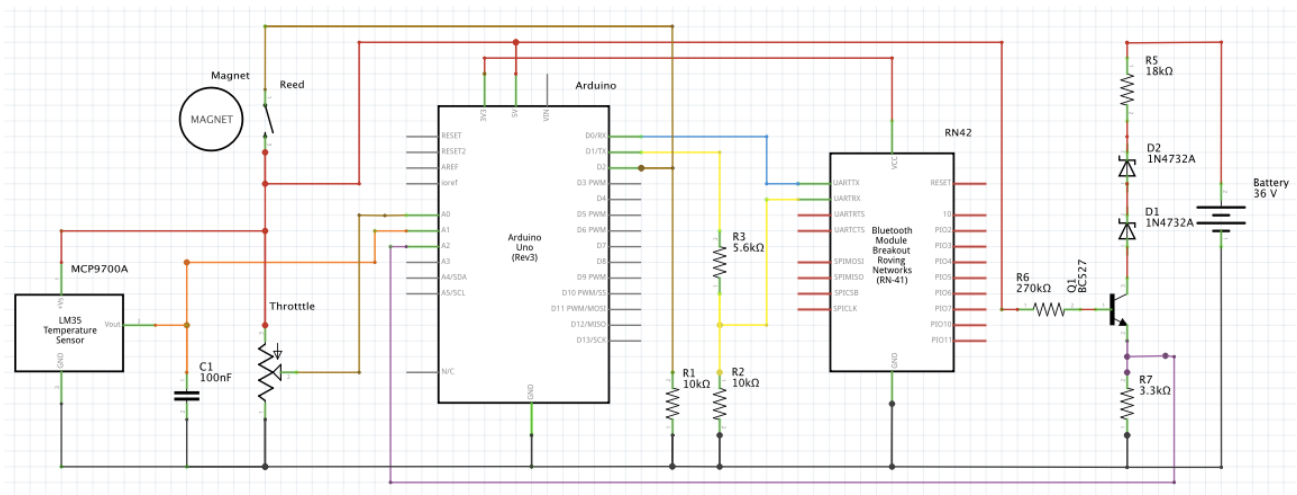


Ilustración 13. Esquemático del circuito final.

El resultado final del circuito completo se puede ver en la *Ilustración 14*, dónde está representado todos los elementos que forman parte del Hardware del proyecto.

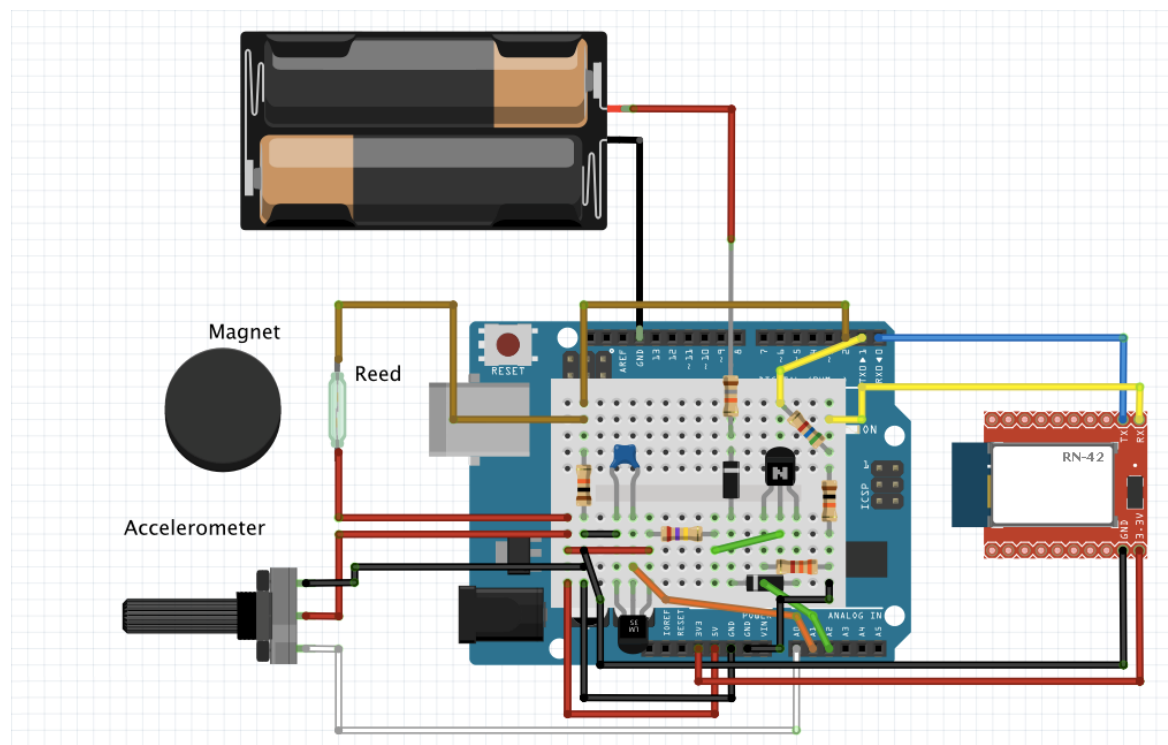


Ilustración 14. Diseño del circuito final en la plataforma Arduino.

3.2.8. Programación del Hardware en Arduino.

A continuación, se van a describir las funciones realizadas por el microprocesador integrado en Arduino. Básicamente, la principal función que realiza es recolectar los parámetros de los diferentes sensores, procesar los datos obtenidos y enviarlos para que la aplicación los pueda representar para el usuario.

Para desarrollar la programación de la plataforma Arduino se programaron por separado cada uno de los circuitos sensores explicados anteriormente en este apartado. Seguidamente, se describirá el proceso de codificación de cada función que realiza.

Odómetro

La función de odómetro está relacionada con el circuito del sensor *reed*. Esta función obtiene los parámetros de conducción por parte del usuario como son: velocidad actual, velocidad media, distancia total recorrida, distancia parcial recorrida y tiempo de uso parcial.

Primero, se implementó un tiempo de referencia con la función *millis()* de Arduino que devuelve el número de milisegundos desde que la placa Arduino empezó a ejecutar el programa. Después, monitorizando la entrada digital reservada para el sensor *reed*, se obtenía el instante en el cual se produjo y se le asignó un contador asociado. Entonces, con estas dos variables y el diámetro de la rueda motriz del *handbike* Batec se obtuvieron todos los parámetros de conducción antes mencionados. En el caso de los parámetros parciales se almacenaban en diferentes variables.

Estado de la batería

La función estado de la batería devuelve el valor de la capacidad restante de la batería en porcentaje. Para implementarla, primero se establecieron los valores límite para cada punto de porcentaje en base al estudio previo realizado de la curva de descarga de la batería. Seguidamente, se comparan estos límites con el valor obtenido de tensión procedente de la batería y se establece un valor aproximado de la carga actual.

Acelerador

La función de la lectura del estado del acelerador nos proporciona el porcentaje de accionamiento del acelerador y también el consumo instantáneo en Amperios. Para implementar esta función, establecemos el valor mínimo y máximo que podemos obtener de la tensión de control del acelerador y lo dividimos en cien partes. Entonces, con un simple bucle obtenemos el parámetro. En el caso del consumo sabemos que éste es proporcional al porcentaje de accionamiento del acelerador, por tanto, una vez obtenido uno tenemos el otro.

Temperatura

La función temperatura obtiene la temperatura ambiente a partir de la tensión de salida del circuito integrado comentado en el sub-apartado anterior. Una vez obtenemos ese voltaje, mediante el factor de conversión que nos proporciona el fabricante obtenemos este parámetro en grados Celsius.

Una vez testeados todos los circuitos con su correspondiente código asociado, se integró en un solo programa todos los códigos desarrollados. Además, se implementó una función que permitía guardar los datos más relevantes en la memoria no volátil de Arduino o EEPROM. Esta función carga los datos al iniciar el programa y los guarda cada vez que detecta que el vehículo se encuentra parado, dato que proporciona la inactividad del sensor *reed*.

Para el envío de los parámetros comentados antes al módulo Bluetooth se utilizó el lenguaje JSON, así la aplicación reconocería rápidamente la variable con su valor asociado de una manera fácil y rápida.

Todos los archivos correspondientes al código desarrollado para programar la placa Arduino se encuentran en el archivo comprimido adjunto a esta memoria.

4. Resultados

En este apartado se mostrarán los resultados obtenidos del proyecto. Se mostrarán los resultados obtenidos por la aplicación Batec Connected, con una explicación de cada página y sus principales funciones. Posteriormente, se dará un listado de algunas funciones de utilidad que ofrece la aplicación.

4.1. Batec Connected: Home Page.

La *Home Page* de la aplicación es la página que nos encontraremos nada más abrir la aplicación, *Ilustración 15*. En ella podemos ver cinco elementos esenciales, cuatro de ellos corresponden al acceso de las siguientes páginas, y el último corresponde al inicio de la función de emparejamiento de la conexión Bluetooth entre el terminal móvil y el prototipo.

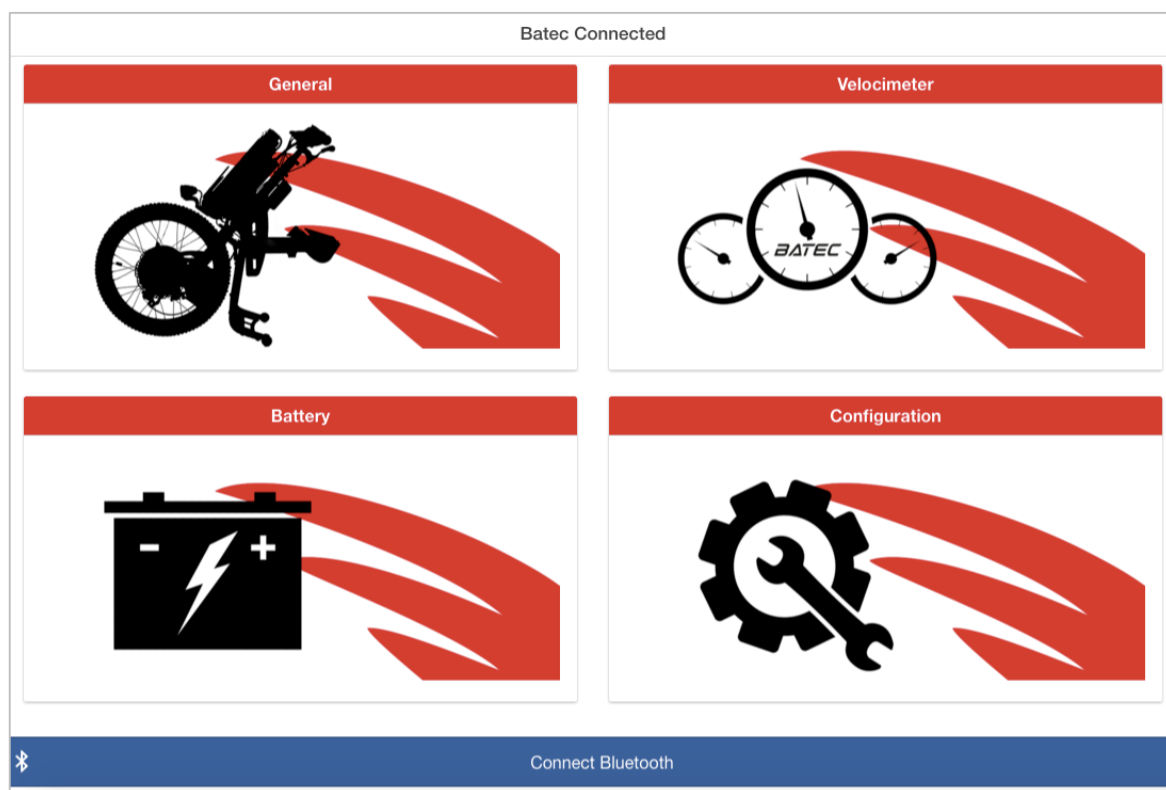


Ilustración 15. Batec Connected Home Page

4.2. Batec Connected: General Page.

La página *General Page* es la página donde un usuario puede tener los datos genéricos más relevantes a su disposición con solo una simple ojeada, *Ilustración 16*. En ella podemos ver cuatro principales zonas.

Primero, en la parte superior izquierda, tenemos el icono para regresar a la *Home Page*. En la parte izquierda tenemos una representación del estado de la batería en forma de pila que va reduciendo la parte en color a medida que la batería gasta su capacidad. En la parte superior derecha, tenemos un velocímetro digital que muestra la velocidad actual

del usuario. Finalmente, si deslizamos el dedo hacia los lados en la parte inferior derecha se pueden visualizar los siguientes parámetros:

- Fecha actual en formato, día de la semana - mes - año.
- Hora actual en formato 24h.
- Recorrido total en km o millas.
- Temperatura en grados Celsius o Fahrenheit.

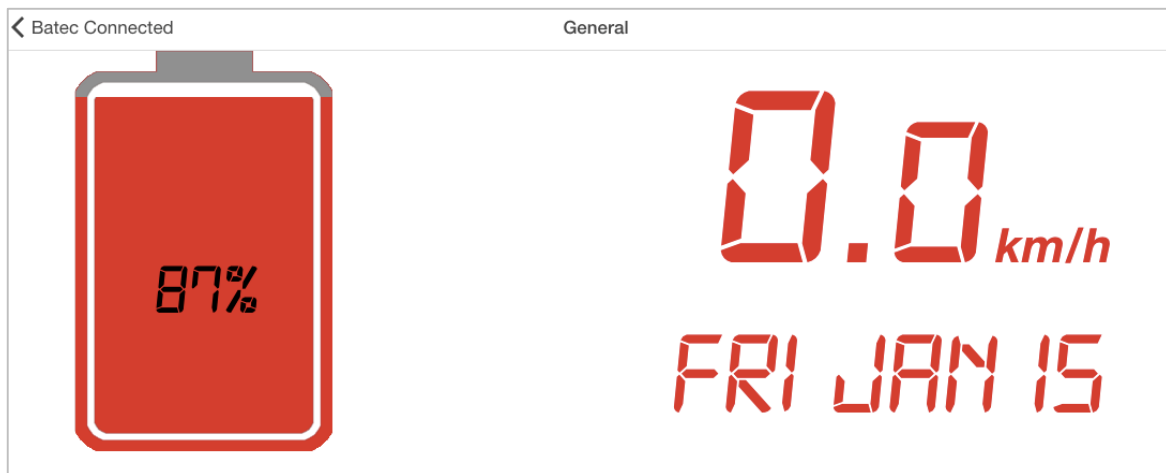


Ilustración 16. Batec Connected General Page

4.3. **Batec Connected: Velocimeter Page.**

La página *Velocimeter Page* es la página donde el usuario puede consultar los datos referentes a la conducción, como se puede ver en el ejemplo de la *Ilustración 17*.

En ella podemos ver cuatro principales zonas. Como antes, tenemos el botón para dirigirnos a la *Home Page* y el velocímetro digital, pero en este caso a la derecha de la página se puede ver una barra semicircular que se llena en función del estado del acelerador con su valor en tanto por ciento en el centro. También tenemos una zona que deslizando el dedo hacia los lados se pueden visualizar los siguientes parámetros:

- Hora actual en formato 24h.
- Velocidad media en km/h o mph.
- Recorrido parcial en km o millas.
- El tiempo de uso parcial, referenciado con el recorrido parcial.
- Recorrido total en km o millas.
- Temperatura en grados Celsius o Fahrenheit.



Ilustración 17. Batec Connected Velocimeter Page

4.4. Batec Connected: Battery Page.

La página *Battery Page* es la página donde el usuario puede consultar los datos referentes al consumo de la batería, como se puede ver en la *Ilustración 18*.

En ella podemos ver también, cuatro zonas principales. Tenemos el botón para dirigirnos a la *Home Page* y la representación del estado de la batería como en la *General Page*. En la parte superior derecha podemos ver el consumo instantáneo en Amperios del *handbike* Batec. Como en las páginas anteriores, tenemos una zona que deslizando el dedo hacia los lados se pueden visualizar los siguientes parámetros:

- Hora actual en formato 24h.
- Consumo medio en Amperios.
- Estimación de la distancia restante con el porcentaje de batería actual.
- Temperatura en grados Celsius o Fahrenheit.

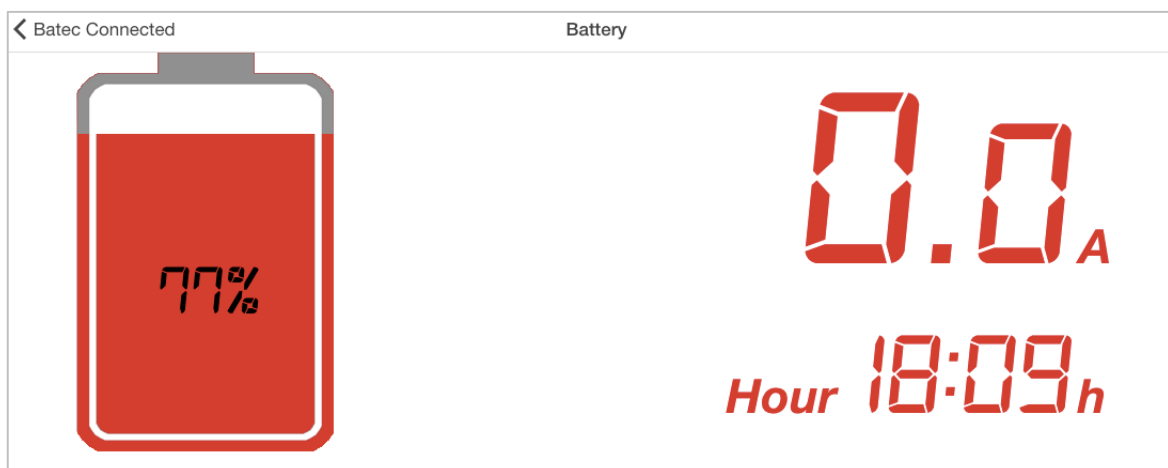


Ilustración 18. Batec Connected Battery Page.

4.5. Batec Connected: Configuration Page.

Finalmente, la última página *Configuration Page* es la página donde se pueden modificar los parámetros de la aplicación. En ella se pueden modificar dos parámetros: El color de toda la aplicación dividido entre las seis gamas de colores disponibles para los *handbikes* Batec y las unidades de medida de distancia y temperatura. En la *Ilustración 19* se puede ver claramente todas las opciones a seleccionar.

Batec Connected		Configuration	
Color Settings			
Batec Red			✓
Batec Blue			
Batec Orange			
Batec Grey			
Batec Rapid			
Batec Purple			
Units Settings			
Length Units in kilometers			<input checked="" type="checkbox"/>
Temperature Units in Celsius			<input checked="" type="checkbox"/>

Ilustración 19. Batec Connected Configuration Page.

4.6. Funciones de utilidad.

A continuación, se presenta un listado con algunas de las funciones que puede que no se aprecien cuándo está en funcionamiento, pero resultan de gran utilidad para el manejo.

- La aplicación reconoce si el dispositivo Bluetooth se encuentra conectado o no. También te avisa en el caso de que no tengas conectado el Bluetooth del teléfono móvil para que lo actives.
- Cuando la aplicación está sincronizada con el dispositivo, ésta evita que la pantalla se apague. Esto resulta muy útil cuando estamos usando el prototipo para no tener que estar constantemente pulsando la pantalla al viajar.
- El dispositivo emite alertas de mantenimiento cuando proceden. Por ejemplo, si el usuario llega a la cifra de 500 kilómetros recorridos, le indica que es aconsejable cambiar el neumático o en el caso de que la batería se encuentre por debajo de un 15%, manda un aviso de precaución.
- La aplicación dará una estimación en kilómetros o millas restantes según los parámetros de conducción del usuario, velocidad media, tiempo activo, consumo medio, etcétera.

- Los datos de cada Batec son independientes de cada aplicación por tanto tan solo enlazando el terminal móvil con el dispositivo se muestran los datos del Batec enlazado. Esto es debido a que todos los datos indispensables para cada unidad se almacenan en la memoria del dispositivo, y la aplicación solo representa los datos que recibe.
- Aunque no se tenga conectado el terminal móvil al dispositivo mientras estas utilizando el Batec, el dispositivo seguirá actualizando todos los datos de conducción que realices, lo único que no serán visibles. Una vez se establezca una nueva conexión los datos se actualizarán de inmediato al terminal móvil.
- Siempre estarán disponibles los últimos datos enlazados con el dispositivo porque estos se almacenan en la memoria de la aplicación. Así, siempre podemos consultar que porcentaje de batería nos resta en nuestro vehículo.
- Cada vez que se detecta que el *handbike* Batec se ha parado, el dispositivo guarda automáticamente los valores obtenidos durante el transcurso del viaje.

5. Presupuesto

En este apartado se detallan los costes de todo el proyecto, se incluyen costes en componentes, en herramientas informáticas y en recursos humanos. En la siguiente tabla podemos ver el precio total del proyecto con el desglose de cada apartado comentado anteriormente.

Componentes	Unidades	Coste unidad	Total
Arduino UNO	1	20.00 €	20.00 €
Arduino Shield MiniProtoboard	1	9.05 €	9.05 €
MICROCHIP RN42XVP-I/RM BT Module	1	14.87 €	14.87 €
FTDI TTL-232R-3V3	1	18.16 €	18.16 €
MICROCHIP MCP 9700A-E/TO Thermistor	1	0.39 €	0.39 €
MULTICOMP BC547B Bipolar Transistor	1	0.23 €	0.23 €
SIGMA BC 5.12 Bike Computer	1	29.77 €	29.77 €
EZEE Twist Throttle	1	38.07 €	38.07 €
Variedad de resistencias	6	0.05 €	0.30 €
Variedad de condensadores	1	0.12 €	0.12 €
Variedad de cables M/M	1	2.48 €	2.48 €
Diodo Zener	2	0.11 €	0.22 €
		<i>Subtotal</i>	133.66 €
Programación	Unidades	Coste unidad	Total
Ordenador	1	450.00 €	450.00 €
Programa Arduino	1	GRATUITO	- €
Microsoft Visual Studio Code	1	GRATUITO	- €
		<i>Subtotal</i>	450.00 €
Recursos Humanos	Horas	Coste Hora	Total
Diseño	126	10.00 €	1,255.80 €
Prototipo	420	10.00 €	4,204.20 €
Total Recursos Humanos	546	10.00 €	5,460.00 €
			TOTAL
			6,043.66 €

Tabla 8. Presupuesto Batec Connected.



Ilustración 20. Distribución del presupuesto de Batec Connected

Como se puede comprobar en el presupuesto del proyecto arriba detallado, aunque el objetivo del mismo es el desarrollo de un producto físico para su venta en el mercado, la mayor parte del coste está basado en los recursos humanos necesarios para el desarrollo de la aplicación.

Esto implica que la amortización y rentabilidad del mismo serán fácilmente alcanzables dado el reducido valor de los materiales que lo componen, de forma que con pocas unidades de venta a un precio altamente competitivo, el proyecto obtendrá una rápida rentabilidad.

6. Desarrollo futuro y conclusiones

En este apartado se presentan distintas mejoras para este proyecto divididas en mejoras de Software y Hardware. Posteriormente se expondrán las conclusiones finales del proyecto.

6.1. Mejoras de Software.

Este prototipo tiene mucho potencial a nivel de Software. Seguidamente se van a argumentar dos posibles funciones adicionales que mejorarían en gran medida la aplicación móvil y sería mucho más atractiva para el usuario.

La integración de Google Maps.

Con la integración de la API de Google Maps, se conseguiría una gran cantidad de funciones adicionales, que van desde una consulta de un recorrido hacia un lugar en concreto, hasta la función GPS que se ejecutara mientras conduces tu Batec sin salir de la aplicación Batec Connected. Además, se podría estimar si con la carga restante de la batería actual sería viable el recorrido seleccionado en base a los parámetros de conducción que el dispositivo almacena.

Integración del Marketing de Batec

Sería posible añadir una función que permita al usuario ver en tiempo real todos los handbikes Batec cercanos a su ubicación, con el fin de promover las relaciones sociales entre los propios usuarios.

También se podría sincronizar la aplicación con las redes sociales y el *newsletter* de Batec Mobility S.L. para que el usuario pueda saber de una forma más sencilla e inmediata toda la información relevante del producto Batec.

6.2. Mejoras de Hardware.

A continuación, se argumentan posibles mejoras en el Hardware del dispositivo que podrían ser aplicadas en la versión final del producto para poder ampliar sus funcionalidades o mejorar las existentes.

Mayor precisión en la lectura del estado de carga de la batería.

Un gran avance, consistiría en la mejora de la precisión sobre la capacidad y el porcentaje actual que se extrae de la tensión de la batería del Batec. Gracias a que la batería utilizada por Batec en sus handbikes tiene un puerto de comunicaciones I2C, el cual no he tenido acceso desde un principio, con el cuál se puede saber los datos exactos que maneja el BMS tales como la tensión, temperatura y capacidad restante de cada una de sus celdas. Entonces se le encomendaría al fabricante de las baterías que habilitase una salida de este puerto de comunicaciones I2C para poder extraer los datos deseados de la batería para poder procesarlos con el microprocesador.

Mejora en la memoria del procesador

Como la memoria EEPROM tiene unos ciclos de escritura limitados, entre unos 100.000 y 1.000.000, resultaría un problema en casos donde haya un uso muy intensivo y longevo del Batec. Por este motivo, en la versión final utilizaríamos una memoria del tipo FRAM en vez de EEPROM porque no tenemos límite de ciclos de escritura de datos.

Implementar un sistema de detección de posibles averías.

El *handbike* Batec incorpora en su electrónica un sistema de alerta de posibles averías, sin embargo, este sistema no es accesible para el usuario. Entonces, obteniendo esa información y mostrándola al usuario para que el usuario tenga conocimiento de inmediato sobre qué elemento electrónico le falla en caso de avería. Con esta implementación proporcionaríamos al usuario una mayor seguridad y confianza hacia el producto agilizando su reparación cuando fuera necesaria.

Conclusiones

Las conclusiones que podemos sacar de este proyecto de tesis es que se trata de un dispositivo que en su versión final será un producto viable para su industrialización, implementación y posterior lanzamiento en el mercado actual de Batec como una mejora de innovación del producto base.

El desarrollo de este proyecto ha sido una experiencia enriquecedora para mí porque he podido aplicar lo que más me apasiona en la vida, el poder ayudar a mejorar, ni que sea un poco, la independencia de las personas que luchan contra una capacidad de movilidad reducida. Además, con este proyecto he podido combinar los dos mundos que más me apasionan, la revolución de la movilidad y las tecnologías de la información y la comunicación.

Bibliografía

- [1]. Arduino. (s.f.). *Arduino Forum*. Recuperado el 31 de Noviembre de 2015, de Arduino: <http://forum.arduino.cc/>
- [2]. BMZ. (19 de Diciembre de 2011). Lithium Ion Manganese Cell BM18650Z3. Karlstein, Bayern, Germany.
- [3]. CADEX. (22 de Julio de 2015). *How to Measure State-of-Charge*. Recuperado el 30 de Noviembre de 2015, de Battery University: http://batteryuniversity.com/learn/article/how_to_measure_state_of_charge
- [4]. CodePen. (s.f.). *Ionic CodePen*. Recuperado el 28 de Octubre de 28, de CodePen: A playground for the front end web: <http://codepen.io/ionic/>
- [5]. Coleman, D. (3 de Diciembre de 2015). *Cordova Plugin for Serial Communication over Bluetooth*. Recuperado el 22 de Diciembre de 2015, de GitHub: <https://github.com/don/BluetoothSerial>
- [6]. MICROCHIP. (26 de Marzo de 2009). Low-Power Linear Active Thermistor™ ICs. Chandler and Tempe, Arizona, EEUU.
- [7]. MIT. (s.f.). *AngularJS: Development Guide*. Recuperado el 13 de Octubre de 2015, de Guide to AngularJS Documentation: <https://docs.angularjs.org/guide>
- [8]. MIT. (s.f.). *Ionic Framework*. Recuperado el 30 de Septiembre de 2015, de Ionic: Advanced HTML5 Hybrid Mobile App Framework: <http://ionicframework.com/>
- [9]. Refsnes Data. (s.f.). *w3Schools Online Web Tutorials*. Recuperado el 9 de Septiembre de 2015, de w3schools.com: The worlds largest web developer site: <http://www.w3schools.com/>
- [10]. Robin Networks, Inc. (12 de Marzo de 2012). RN41XV & RN42XV Bluetooth Module. San José, California, EEUU.
- [11]. Wei, F. (16 de Noviembre de 2015). *Crosswalk WebView Cordova Plugin*. Recuperado el 9 de Enero de 2016, de GitHub: <https://github.com/crosswalk-project/cordova-plugin-crosswalk-webview>

Glosario

API – Application Programming Interface.

BJT – Bipolar Junction Transistor

BLE – Bluetooth Low Energy.

BMS – Battery Management System.

CSS – Cascading Style Sheets.

EEPROM – Electrically Erasable Programmable Read-Only Memory.

FRAM – Ferroelectric Random Access Memory.

GPS – Global Positioning System

HTML5 – HyperText Markup Language 5.

MVC – Model View Controller.

SASS – Syntactically Awesome Style Sheets.

SDK – Software Development Kit.

W3C – World Wide Web Consortium.